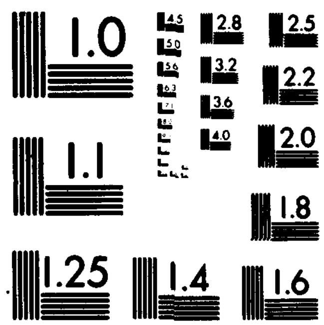
AD-A087 357 MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR COMPUTE--ETC F/G 9/2
PROGRAMS FOR DISTRIBUTED COMPUTING: THE CALENDAR APPLICATION.(U)
JUL 80 I GREIF N00014-75-C-0661
UNCLASSIFIED MIT/LCS/TM-168 NL

END
DATE
FILMED
9—80
DTIC



MICROCOPY RESOLUTION TEST CHART

LEVEL II

12

LABORATORY FOR COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MIT/LCS/TM-168

PROGRAMS FOR DISTRIBUTED COMPUTING: THE CALENDAR APPLICATION

Irene Greif

July 1980

This research was supported by the Advanced Research Projects Agency of Defense and was monitored by the Office of Naval Research under Contract No. N00014-75-C-0661

545 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS 02139

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 14 MIT/LCS/TM-168	2. GOVT ACCESSION NO. AD-A087 357	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 Programs for Distributed [redacted] Computing: The Calendar Application		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER MIT/LCS/TM-168
7. AUTHOR(s) 10 Irene Greif		8. CONTRACT OR GRANT NUMBER(s) 15 N00014-75-C-0661
9. PERFORMING ORGANIZATION NAME AND ADDRESS MIT/Laboratory for Computer Science 545 Technology Square Cambridge, MA 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 12 11
11. CONTROLLING OFFICE NAME AND ADDRESS ARPA/Department of Defense 1400 Wilson Boulevard Arlington, VA 22209		12. REPORT DATE 11 July 1980
		13. NUMBER OF PAGES 10
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) ONR/Department of the Navy Information Systems Program Arlington, VA 22217		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The calendar application involves a wide range of issues in distributed computing, from implementation of distributed data bases to design of a user interface that will enable the user to comprehend the complex distributed environment in which he is working. This memo summarizes current status of design and implementation of calendars. Sections 2 and 3 are taken from a progress report of March 1980 and section 4 is an update to that report including current status plans. We began our design of calendar systems with the intention of focussing on implementation issues, particularly those of

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

409648 JOB

20. communication, sharing of data, and the use of forms as the standard interface among modules. To assure variety in our implementation experience, we generated alternative designs of calendars. Implementations of several of them are now underway. During later design phases we became more interested in the functionality of the calendar and its user interface. To some extent the use of forms as the communication medium shapes the human interface as well as the process-to-process interface. Also, some aspects of the functionality, most notably the notion of "tentative meeting" have arisen from analysis of the distributed implementation. Section two is a report on the functionality of the calendars we are building. Section three is about the distributed implementation.

Programs for Distributed Computing: The Calendar Application

by

Irene Greif

18 June 1980

This research was supported by the Advanced Research Projects Agency of Defense and was monitored by the Office of Naval Research under Contract Number N00014-75-C-0661.

1. Introduction

The calendar application involves a wide range of issues in distributed computing, from implementation of distributed data bases to design of a user interface that will enable the user to comprehend the complex distributed environment in which he is working. This memo summarizes current status of design and implementation of calendars. Sections 2 and 3 are taken from a progress report of March 1980 and section 4 is an update to that report including current status and plans.

We began our design of calendar systems with the intention of focussing on implementation issues, particularly those of communication, sharing of data, and the use of forms as the standard interface among modules. To assure variety in our implementation experience, we generated alternative designs of calendars. Implementations of several of them are now underway.

During later design phases we became more interested in the functionality of the calendar and its user interface. To some extent the use of forms as the communication medium shapes the human interface as well as the process-to-process interface. Also, some aspects of the functionality, most notably the notion of "tentative meeting" have arisen from analysis of the distributed implementation.[1] Section two is a report on the functionality of the calendars we are building. Section three is about the distributed implementation.

2. The Functionality of the Calendar

There are two kinds of calendars that we have been designing -- personal calendars and public "resource scheduling" calendars.

2.1. The Personal Calendar

The personal calendar can be used for keeping track of appointments, meetings, holidays, etc. The calendar can be displayed in several ways showing either a summary of the week, a list of appointments on a day, or a diagram of the day showing blocks of free and reserved time. The main operations are "appt" to make an appointment, "cancel" to cancel one, and various display commands. A description of the commands will be typed out in response to a question mark.

One can attempt to make an appointment at any time. If there is a conflict with another

[1] Leslie Lamport inspired this line of thought when he insisted on knowing what my calendar did, and started to help me specify the meaning of a call for a meeting.

appointment, the calendar reports this fact. If not, the appointment will be made. Appointments are recorded at a particular time with a few keywords to indicate the purpose.

2.2. A Calendar for a Conference Room

The Conference Room Calendar is similar to the personal calendar in that time slots can be reserved and cancelled. This program is meant to support the reserving of time in one of the conference rooms in our laboratory. These rooms are generally used for seminars the scheduling of which may involve coordination among several people and resources. Since a seminar generally has a host who is responsible for the reservation, the host's name is listed in the calendar display as the keyword for the appointment. In addition, there is a form on file for each appointment. The form contains information about the seminar such as the speaker's name, the title of his talk and whether there will be refreshments. These forms can be active, in which case they may trigger communication with other calendars (such as the calendar for the person who sets up the coffee pot in time for scheduled refreshments).[2]

2.3. Coordination Between Personal Calendars

A personal calendar can try to call a meeting. The desired length of the meeting, a set of possible times and a list of participants must be specified in the request. The calendar system will try to find a time at which the meeting can be held and will then notify all participants.

For meetings that are called very far in advance of the time at which they will be held, the meeting can be considered to be tentatively scheduled. A scheduler will keep track of several possible times at which the meeting can be held. A second meeting is considered to conflict only if scheduling it (and therefore removing its time slot from the set of times tentatively reserved for the original meeting) would reduce the set of possible times to less than one. If the second meeting is scheduled, the set of available times for the first meeting is simply reduced. Shortly before the date of the meeting a single time is chosen for the meeting. This can occur either at a "commit" time specified in the call for the meeting or by an explicit request to commit. A caller could specify that he wants a meeting the week of March 10th and that it should be definitely scheduled by March 3rd. Thus the caller can be sure

---

[2] Simple single process versions of both the personal calendar and the 512 calendar have been implemented. The personal calendar can be run on XX by copying <greif>pcal.exe and <greif> _ xfile.pcal and then typing pcal at exec. For the 512 calendar, copy <greif>512.exe and <greif> _ xfile.512 and type 512. No guarantees are made about the performance or consistency with the description in this paper. The programs will leave some files named "your-name.cal.1", "512.cal.1", and "512.forms.1" in your directory.

that the meeting will appear on his calendar with sufficient advance notice for planning. If the meeting is committed to a single time too soon, it is quite likely that some participant will have to cancel in order to meet a higher priority commitment that arises later. This would require rescheduling, rather than the simple reduction in the set of tentative times.

2.4. Making Requests -- The User Interface

As mentioned above, in the case of seminars we keep a file of forms containing information which does not necessarily appear on the calendar display. For uniformity we implement all interactions with the calendar in terms of forms so that even a reservation for the personal calendar is considered to be accomplished by filling in a form -- a "request for reservation" form. The fields are date, start time, end time and keywords. A cancellation form requires only a date and start time. Requests for displaying parts of the calendar are made by filling in the date.

The user need not know what information is required for a particular request -- the form itself will prompt him once he specifies the kind of request he wishes to make.[3] This is a kind of active form in which the filling in of one field (the type of request field) triggers the asking of further questions -- the form only asks for the fields that contain information relevant to the particular request. Changes to entries can be made by entering edit mode. In this mode the system asks which field is to be modified and for each field, displays the old contents of the field and accepts a new entry. There is currently a distinction made between fields that can always be changed (e.g. title of a talk) and those which require rescheduling (e.g., start time). To change the latter one must enter "reschedule" mode.

As we extend the functionality of the calendar we expect that sometimes filling in one form will trigger filling in an auxiliary form. This could arise, for example, when calling a meeting that has a room requirement. The room requirement implies that the seminar request form must be filled in and sent to the conference room calendar.

Most request forms are disposed of as soon as the request is fulfilled. This is not true of the seminar request forms, since that information may have to be referred to (and possibly even modified) later. These forms are kept on file according to date and time of the seminar. It should also be possible to file incomplete forms that are not yet associated with a definite time. For example, if a request conflicts with scheduled meetings, but an alternative time cannot be suggested immediately,

[3] The current implementation of pcal does not prompt for all fields in "appt" requests.

the user might store the form and retrieve it later, avoiding retyping all of the information in order to resubmit the request. A user interface with more sophisticated forms manipulation facilities is being designed by a bachelor's thesis student [Wylen, 1980].

3. Calendars in a Distributed System

Facilities for coordinating a set of calendars are of use in either a centralized or a distributed system. If the system is to be distributed, its implementation will certainly differ from the implementation of a centralized version. We are assuming that in order to coordinate with another calendar a request must be sent to that calendar. That is, there is no central data base that contains information on all calendars and that can be accessed directly by any calendar.[4] Thus from the point of view of anyone who needs information about several people the data is stored in a distributed data base.

Operations other than calls for meetings may depend on data at more than one node. For example, when there are tentative meetings (as described in section 2.3) then while a meeting is "uncommitted" the status of certain time slots on the personal calendars of the participants may depend on the status of the tentative meeting. Thus even if the personal calendars store their data locally they may have to communicate with the tentative meeting in order to find out whether a particular time slot is free. This can cause noticeable delays if a user is at the terminal trying to schedule an appointment in real time. It also raises a question as to how to display the calendar -- should all tentative times for various meetings be shown or should the display show a possible schedule based on information available locally?

Other questions, both of implementation and of functionality, arise:

- How do these data dependencies relate to the dependencies which arise in supporting modular atomic transactions [Reed, 1979]? Are such dependencies at the application level likely to occur in many applications? If so, how can we support their implementation in a programming language for distributed applications?

- Should the caller of the meeting act as the source of information about the tentative meeting? If the tentative meetings are distributed in this way how will scheduling be done if one person is invited to several meetings? Should a central scheduler be invoked to manage meetings? (This latter approach is being explored by a UROP student.)

---

[4] This does not preclude storing all information on a central storage device such as the Swallow Distributed Data Storage System [LCS, 1980] Access to any process' data, e.g. a calendar's data, would still be under that process' control.

- Should chains of tentative meetings be schedulable? (E.g., can I schedule meeting A conditionally depending on the final timing of meeting B?) This may save the time of checking with the tentative meeting about a particular time slot. But then how will the system help me in backing out of meetings when conflicts are later confirmed?

4. Conclusion

Most of the user interface functions described above have been implemented in a single user setting and some simple communicating calendars have been built. Since March the two students mentioned above have completed parts of calendar systems.

Eli Wylen implemented user interface routines that allow the user to chose among different views of the calendar -- week, day, list of appointments, expanded view of a form associated with an appointment. He has also provided some facilities for editing calendars and forms, integrating standard text editing features for moving the cursor around within a field with additional cursor movement facilities for moving around on either the form or the calendar.

Pat O'Donnell, a UROP student, has implemented a scheduler that can handle tentative meetings. The actions of the participants, including the caller of the meeting, are currently all controlled by a single process used for testing the scheduler. However, the scheduler and this testing process are run as separate jobs on XX with all communication through message passing.

We hope to bring up a robust form of the calendar for general use on XX in the near future. The facilities provided would likely include calendars for several conference rooms, a scheduler for tentative meetings, and personal calendars with facilities to support sharing, e.g. between secretary and faculty member. The system will simply provide tools for coordinating the use of distributed calendars under the assumption that people will be involved in end-to-end protocols such as confirmations retransmission when no response is received and decision-making when options arise in scheduling.

5. References

Gifford, D. K., "Weighted Voting for Replicated Data," *Proceedings of the Seventh Symposium on Operating Systems Principles*, Pacific Grove, California, December, 1979, pp. 150-162.

MIT Laboratory for Computer Science Progress Report, 1980.

Reed, D. P., "Implementing Atomic Actions on Decentralized Data," *Preprints for the Seventh*

Symposium on Operating Systems Principles, Pacific Grove, California, December, 1979, pp. 66-74.

Wylen, E., "A Personal Calendar: The Human Computer Interface." Bachelor's Thesis, Expected June, 1980.

OFFICIAL DISTRIBUTION LIST

Defense Technical Information Center
Cameron Station
Alexandria, VA 22314 12 copies

Office of Naval Research
Information Systems Program
Code 437
Arlington, VA 22217
2 copies

Office of Naval Research
Branch Office/Boston
Building 114, Section D
666 Summer Street
Boston, MA 02210
1 copy

Office of Naval Research
Branch Office/Chicago
536 South Clark Street
Chicago, IL 60605
1 copy

Office of Naval Research
Branch Office/Pasadena
1030 East Green Street
Pasadena, CA 91106
1 copy

New York Area
715 Broadway - 5th floor
New York, N. Y. 10003
1 copy

Naval Research Laboratory
Technical Information Division
Code 2627
Washington, D. C. 20375
6 copies

Assistant Chief for Technology
Office of Naval Research
Code 200
Arlington, VA 22217
1 copy

Office of Naval Research
Code 455
Arlington, VA 22217
1 copy

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
(Code RD-1)
Washington, D. C. 20380
1 copy

Office of Naval Research
Code 458
Arlington, VA 22217
1 copy

Naval Ocean Systems Center, Code 91
Headquarters-Computer Sciences &
Simulation Department
San Diego, CA 92152
Mr. Lloyd Z. Maudlin
1 copy

Mr. E. H. Gleissner
Naval Ship Research & Development Center
Computation & Math Department
Bethesda, MD 20084
1 copy

Captain Grace M. Hopper, USNR
NAVDAC-OOH
Department of the Navy
Washingon, D. C. 20374
1 copy

DAT
FILM